Summary of Projects

1. Profit-Loss Calculator
This project provides a simple GUI-based calculator where users can input selling price (SP), market price (MP), and select their role as a buyer or seller. The program then computes whether they have made a profit, incurred a loss, or achieved breakeven. It helps users quickly evaluate transaction outcomes in financial terms.

2. Financial Calculator
This Python-based financial tool performs essential investment computations. It calculates Net Present Value (NPV), Internal Rate of Return (IRR), and Discounted Cash Flows (DCF) based on stock price data. The calculator enables quick investment analysis, making it useful for decision-making in finance.

3. Risk Analysis Tool
This project fetches one year of stock data from Yahoo Finance and computes multiple risk-return metrics. Through a simple Tkinter GUI, users can enter a stock ticker (e.g., AAPL, MSFT) and receive detailed insights on stock returns and associated risks. It is a practical application of risk management concepts in portfolio analysis.

4. Stock Price Forecasting with ARIMA
This project applies the ARIMA (AutoRegressive Integrated Moving Average) model to forecast stock prices. Using real-time data from Yahoo Finance, the system generates numerical predictions and visual plots of future stock performance. A GUI interface enhances usability, making time-series forecasting accessible to users.

5. GARCH Model
The GARCH (Generalized Autoregressive Conditional Heteroskedasticity) model is implemented to estimate and forecast volatility in stock returns. By modeling conditional variance using past errors and variances, the project demonstrates how volatility forecasting supports risk management, option pricing, and Value-at-Risk (VaR) estimation.

6. Monte Carlo Simulation Model
This project applies Monte Carlo methods to simulate possible future stock prices using historical market data. By combining finance concepts, probability modeling, and GUI development, the project demonstrates how randomness can be harnessed to assess the uncertainty of investment outcomes.

7. Modern Portfolio Theory (Extended Project)
This project implements Modern Portfolio Theory (MPT) using Python, real stock data, and simulation techniques. It shows how diversification can minimize risk while maximizing returns. The project calculates annualized returns, volatility, covariance, and performs portfolio optimization. Monte Carlo simulations further illustrate how portfolios can be constructed to optimize the risk-return tradeoff.

Extended Project: Modern Portfolio Theory with Monte Carlo Simulation

The extended project focuses on **Modern Portfolio Theory (MPT)**, a foundational concept in finance developed by Harry Markowitz. MPT emphasizes diversification, demonstrating that investors can maximize returns for a given level of risk (or minimize risk for a desired level of return) by carefully selecting a mix of assets.

In this project, historical stock data from multiple companies (e.g., AAPL, MSFT, GOOGL, AMZN, TSLA) was collected using Yahoo Finance. Key financial metrics such as log returns, annualized returns, volatility, and covariance were computed. Using this data, the project applied Monte Carlo simulations to generate thousands of random portfolios, each with varying asset weights.

The simulation results were plotted to visualize the **Efficient Frontier**, showing the tradeoff between risk (volatility) and expected return. From this analysis, the project identified:

- The **Maximum Sharpe Ratio Portfolio**, which offers the best risk-adjusted return.
- The **Minimum Volatility Portfolio**, which minimizes risk exposure.

This project not only implements MPT but also integrates advanced optimization and visualization techniques, bridging theoretical finance concepts with practical coding applications.

Enhancements

Compared to standard implementations, this project was extended with the following enhancements:

- **Monte Carlo Simulation**: Generated 20,000 random portfolios to explore the full range of possible risk-return combinations.
- **Sharpe Ratio Calculation**: Incorporated the risk-free rate into portfolio evaluation, enabling identification of the optimal risk-adjusted portfolio.
- **Efficient Frontier Plot**: Visualized all simulated portfolios along with highlighted optimal portfolios, providing a clear graphical understanding of diversification benefits.
- **Integration of Multiple Assets**: Used a diverse set of stocks from different industries to reflect real-world portfolio construction.
- **Clear Annotation & GUI-ready Structure**: Designed the output with detailed labels and markers for easy interpretation.

Key Takeaways

1. **Understanding of Financial Concepts**

- Gained practical knowledge of Modern Portfolio Theory, Monte Carlo methods, ARIMA forecasting, and volatility modeling with GARCH.
- Understood how risk, return, and diversification interact in financial decision-making.

2. **Technical & Coding Skills**
   - Strengthened Python programming in finance using libraries such as `numpy`, `pandas`, `matplotlib`, `yfinance`, and `scipy`.
   - Implemented advanced statistical models (ARIMA, GARCH) and simulation techniques (Monte Carlo).
   - Built interactive GUIs with Tkinter, enhancing usability.
3. **Visualization & Analysis**
   - Developed skills in data visualization for financial insights (efficient frontier, risk-return scatter plots, time-series forecasting).
   - Learned how to annotate and present results clearly for decision-making.
4. **Practical Application**
   - Applied theoretical finance models to real-world stock data.
   - Built a portfolio optimization tool that can be extended for research, investment, or academic use.